Numerical Optimization

WEEK 2

Course Outline

- Taylor Series
- Optimizing Functions with one variable
  - First order necessary conditions
  - Second order necessary conditions
  - Second order sufficient conditions
- Optimizing Functions with multiple variables
  - First order necessary
  - Second order necessary and sufficient conditions
- Gradient Descent
- Newton’s method

Taylor Series

Taylor Series are used to approximate a function by help of its derivative values

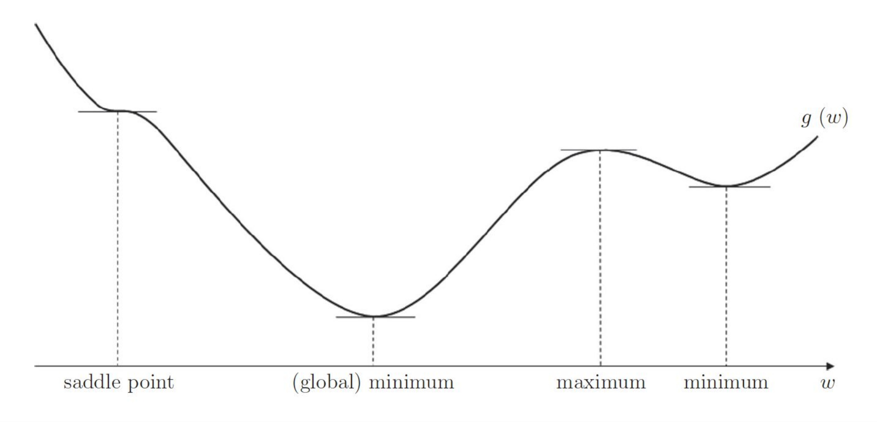
$$\sum_{n=0}^{\infty} \frac{f^{(n)}(a)}{n!}(x-a)^n$$

As n (degree) increases, the error between Taylor Series and original function value decreases.

Functions with one variable

First order necessary conditions $f'(x^*) = 0$

The point x^* in this case can be local maximum, local minimum or a inflection point(saddle point)



Source: Machine Learning Refined TextBook

Example

Ex.1

$$f(x) = x^2 - 3x + 2$$

$$f' = 2x - 3$$

$$2x - 3 = 0 \longrightarrow x = \frac{3}{2}$$

Ex.2

$$g(w) = w^{\frac{3}{2}} - 3w$$

$$g' = \frac{3}{2}\sqrt{w} - 3$$

$$\frac{3}{2}\sqrt{w} - 3 = 0 \longrightarrow w = 4$$

Functions with one variable

Second order necessary conditions for a local minimum

$$f''(x^*) \geq 0$$

Second Order Sufficient conditions for a local minimum

$$f''(x^*) > 0$$

Functions with one variable

Previous Ex.1

$$f(x) = x^2 - 3x + 2$$

$$g(w) = w^{\frac{3}{2}} - 3w$$

$$f' = 2x - 3$$

$$f''(3/2) = 2 > 0$$

$$g' = \frac{3}{2}\sqrt{w} - 3$$

$$g'' = \frac{3}{4\sqrt{w}}$$

$$g''(4) = \frac{3}{4\sqrt{4}} = \frac{3}{8} > 0$$

Functions with multiple variables

Gradient of a function

$$\nabla f(x,y) = i\frac{\partial f}{\partial x} + j\frac{\partial f}{\partial y}$$

$$\nabla f(x,y) = [\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}]^T$$

Hessian Matrix

$$\nabla^2 f = \mathbf{H} = \begin{bmatrix} \frac{\partial^2 f}{\partial x \partial x} & \frac{\partial^2 f}{\partial x \partial y} \\ \frac{\partial^2 f}{\partial y \partial x} & \frac{\partial^2 f}{\partial y \partial y} \end{bmatrix}$$

Functions with multiple variables

First order necessary conditions

$$\nabla f = 0$$

Same with one variable, the point $x^* = (x_1, x_2, x_3, x_4, \ldots)$ can be local maximum, local minimum and inflection point (saddle point)

Functions with multiple variables

Second order necessary and sufficient conditions

We cannot compare a scalar with a 2 dimensional Hessian matrix.

We can using eigenvalues to check if the Hessian matrix satisfies the conditions.

Functions with multiple variables

$\mathbf{H}$ is positive definite if and only if $\lambda_i > 0, \forall i = 1...n$

$\mathbf{H}$ is negative definite if and only if $\lambda_i < 0, \forall i = 1...n$

$\mathbf{H}$ is positive semidefinite if and only if $\lambda_i \geq 0, \forall i = 1...n$

$\mathbf{H}$ is negative semidefinite if and only if $\lambda_i \leq 0, \forall i = 1...n$

Functions with multiple variables

Given a Hessian Matrix;

Second order **necessary conditions for local minimum** Hessian matrix should be **positive semidefinite** matrix

Second order **necessary conditions for local maximum** Hessian matrix should be **negative semidefinite** matrix

Second order **sufficient conditions for local minimum** Hessian matrix should be **positive definite** matrix

Second order **sufficient conditions for local maximum** Hessian matrix should be **negative definite** matrix

Functions with multiple variables

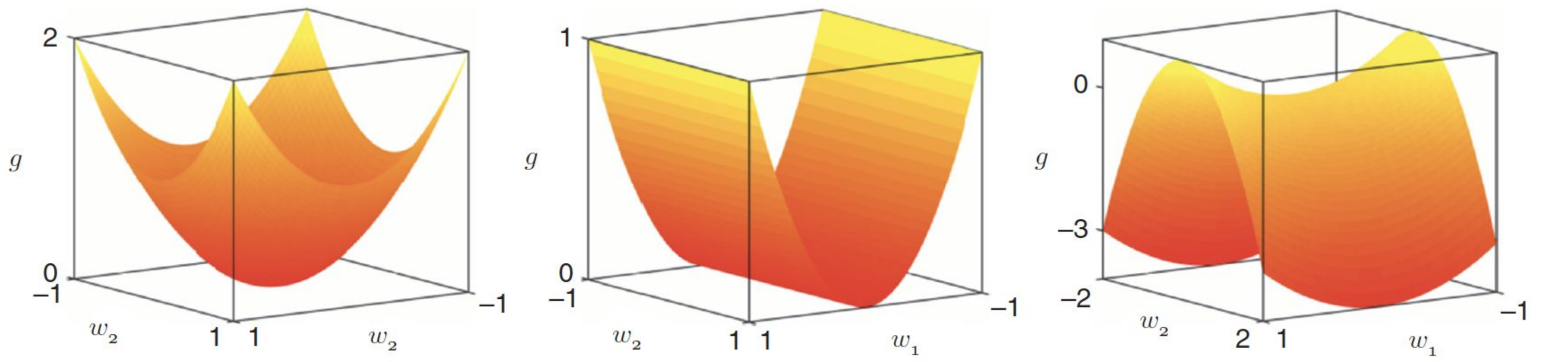
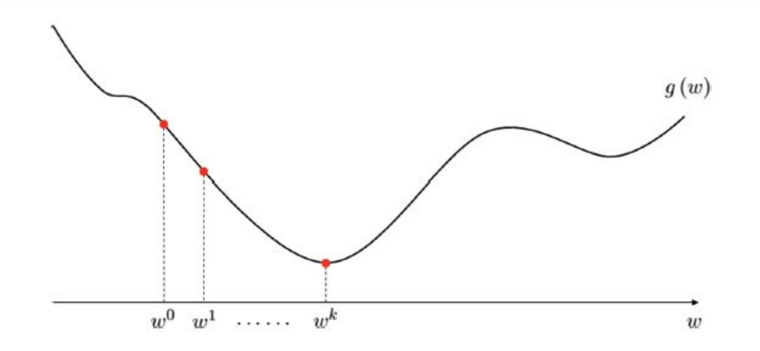


Fig. 2.4 Three quadratic functions of the form $g(\mathbf{w}) = \frac{1}{2}\mathbf{w}^T\mathbf{Q}\mathbf{w} + \mathbf{r}^T\mathbf{w} + d$ generated by different instances of matrix $\mathbf{Q}$ in Example 2.3. In all three cases $\mathbf{r} = \mathbf{0}_{2\times 1}$ and $d = 0$. As can be visually verified, only the first two functions are convex. The last "saddle-looking" function on the right has a saddle point at zero!

Source: Machine Learning Refined TextBook

Pseudo optimization algorithm

1. Start optimization by selecting an initial point w^0
2. Update the parameter w^k depending on w^{k-1}
3. Repeat step 2 until some stopping criteria



Source: Machine Learning Refined TextBook

Stopping Criteria

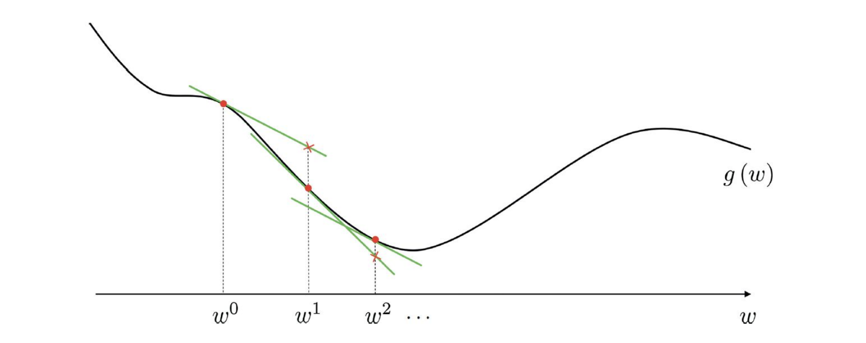
1. When pre-specified number of iterations are complete.
2. When the gradient small enough compared to a pre-specified threshold $\varepsilon>0$.

Gradient Descent Algorithm

k = 1

Repeat until stopping conditions met;

1. $$w^k = w^{k-1} - \alpha * \nabla g(w^{k-1})$$
2. k = k+1



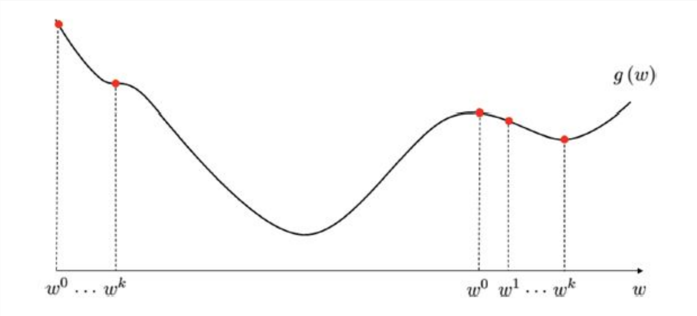
Source: Machine Learning Refined TextBook

Advantages

- Space Efficient
- Always converges in a convex function

Disadvantages

- Convergence depends on initial point
- Chance of getting stuck at inflection point
- Choosing appropriate learning rate(α)



Source: Machine Learning Refined TextBook

Newton Method Formulation

$$f(w) = g(w^0) + \nabla g(w^0)^T (w - w^0) + \frac{1}{2}(w - w^0)^T \nabla^2 g(w^0)(w - w^0)$$

Suppose we make second order approximation at point w^0.

We want to find a minimum/maximum of that approximation, and update our parameter w with the point that is minimum/maximum for our approximation.

$$\nabla^2 g(w^0) w = \nabla^2 g(w^0) w^0 - \nabla g(w^0)$$

Finally if we multiply each side with inverse of the Hessian (second order derivative), we obtain;

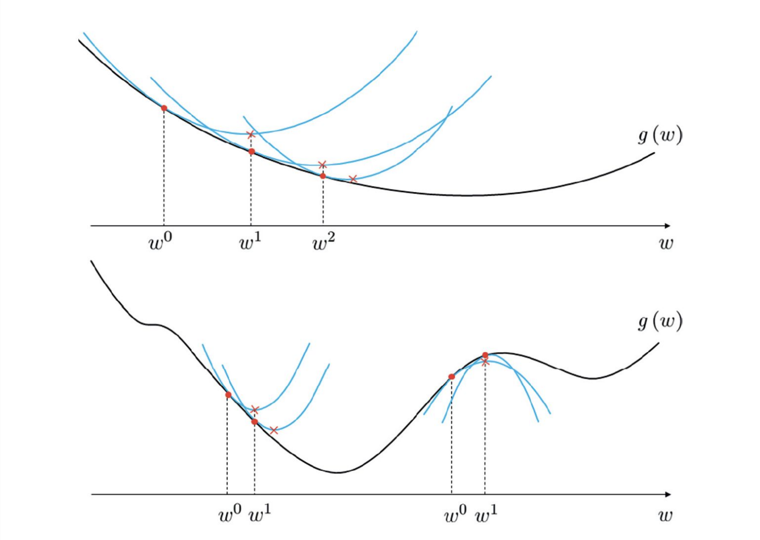
$$w^k = w^{k-1} - [\nabla^2 g(w^{k-1})]^{-1} \nabla g(w^{k-1})$$

Newton Method Algorithm

k=1

Repeat until stopping criteria met;

1. $\nabla^2 g(w^{k-1}) w^k = \nabla^2 g(w^{k-1}) w^{k-1} - \nabla g(w^{k-1})$ for w^k
2. k = k + 1



Source: Machine Learning Refined TextBook

Newton Method

Advantages

- Less hyperparameters to worry about
- Precise and fast convergence for convex functions

Disadvantages

- Storing the Hessian and calculating the inverse is inefficient for larger dimensions
- Convergence depends on initial point for a nonconvex function

Questions?